EPiC Series in Computing

Volume 109, 2026, Pages 120–132

Proceedings of EUNIS 2026 Annual Congress

Integrated Emergency and Service Status Communication – Robust and Target-Specific via a Highly Available Status Board Supporting BCM, ISM, and ITSM

Miran Maximilian Mizani[1,2] and Stefan Metzger[1]

[1] Leibniz Supercomputing Center, Garching, GERMANY
[2] Ludwig-Maximilians-Universität München, GERMANY
lastname@lrz.de
https://orcid.org/0000-0002-7307-6757, https://orcid.org/0000-0002-7307-6757

Abstract

Using the example of the Leibniz Supercomputing Centre (LRZ), this article presents an approach for the design, implementation and practical use of a highly available, web-based emergency and service status communication solution that can continue to operate in an emergency regardless of the availability of central IT systems. The technical architecture, which has also proven itself in our own practice, relies entirely on open-source tools and has enhanced LRZ's (cloud) service status reporting.

The value contribution lies in the technical and organizational integration of daily IT service management workflows for incident and maintenance notifications with external communication during emergencies and crises from the perspectives of business continuity and information security management. Target-group–specific messaging capabilities are realized within this framework. Reusable integration variants for processes in the higher education environment, which often comprises heterogeneous and distributed IT operating groups, are presented and practical experiences are discussed.

Keywords: Service Status Reporting, Emergency Communication System, Business Continuity, Incident Notification, Information Security Management, Cloud Status Board

1 Introduction

Every IT service provider – whether a small IT group, a university's central IT department, or a large data center – faces the daily challenge of handling outages, disruptions, or planned changes to IT services. Failures of core services such as networking, authentication, communication platforms, or e-learning systems can significantly impact workflows, research, and teaching activities of a higher education institution (HEI).

In such situations – and especially in emergency or crisis scenarios – it is not only crucial to quickly restore technical functionality, but also to communicate proactively, purposefully and in a way that builds trust with the affected internal and external user groups. This also helps to counteract overwhelming ticket volumes at the service desk, user frustration and potential reputational damage. A robust (external) communication channel thus becomes an emergency-relevant infrastructure component and contributes to the resilience of the organisation.

In view of rising threats and prominent incidents, the authors note growing awareness and concrete efforts within german HEIs to strengthen resilience, adopt Business Continuity Management (BCM) measures, and prepare for emergencies. From our perspective, a robust and largely failure-resistant solution for (unidirectional) emergency communication to internal and external recipients represents an suitable (first) step in BCM and a helpful building block for Cyber Security Incident Response Teams (CSIRTs).

Using the example of the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences in Germany, this article examines the design, implementation and practical use of a highly available, web-based emergency and service status communication page ("status board") using open-source tools, which remains operational even when central IT systems such as storage or authentication services are unavailable.

From BCM (e.g. according to [3]) and Information Security Management (ISM, e.g. according to ISO/IEC 27001 [5]), this work also enhances (cloud) service status reporting within the cloud and IT service management (ITSM) processes of the examined service provider. The added value of the approach lies integrating everyday procedures for incident and maintenance notification (from ITSM, e.g. according to ISO/IEC 20000-1 [4] or ITIL v4 [1]) with emergency and crisis communication from BCM and ISM. This alignment ensures that familiar operational workflows can be maintained during critical events. Building on [6] and incorporating feedback from early adopters within the german HEI environment, the approach is extended to provide integrated, tailored communication for multiple target groups.

During the design phase, particular attention was paid to the specific requirements of the HEI environment, which is characterised by a heterogeneous user groups, partially federated IT structures, often decentralised services and operator groups, and usually diverse recipient groups without central contact persons, such as a large number of departments and students.

The conceptual design addresses underlying use cases and derives corresponding requirements from the perspectives of different announcement authors and recipients (see chap. 2.1). Chapter 2.2 discusses the transition from IT monitoring to service status reporting. For 3 Implementation of the Service Status Board, decisions on technical architecture, integration and organisational processes are presented in a reusable manner. Finally, chap. 4 reflects on experiences and both technical and organizational challenges encountered during the introduction and nearly three years of productive operation.

2 Design of an emergency communication and service status reporting system

"Service Status Reporting" in this paper refers to the proactive notification of disruptions, outages, or planned maintenance by the service provider that perceptibly affects service quality for users, even when contractual service levels are maintained. Service status here is considered in terms of user-perceived availability and overall quality-of-service.

Proactive service status reporting to users not only reflects professional conduct and common best practices among service providers but also prevents a potential flood of tickets during widespread disruptions. This delivers a direct operational benefit to the helpdesk, which is often understaffed in HEI contexts.

Expansion to add capabilities for unidirectional emergency communication

The core idea of our approach is to integrate the communication channels of service status reporting with those used for external (user) communication in emergencies and crisis, both technically and organizationally. This allows the latter to be understood as a form of service sta-

tus reporting for outages with wide-reaching impact. Depending on their severity, interruptions are handled by ITSM or BCM procedures (see fig. 1).

Unlike trivial approaches that switch to entirely new, independent channels in emergencies (e.g., an external hosted website with separate user management) in an emergency (even ad hoc), our method integrates emergency communication channels from CSIRTs, crisis or PR teams (from ISM and BCM) with daily operational procedures for outage and maintenance notifications (from ITSM). This enables both administrators and users to rely on familiar, routine channels even in emergency situations or crises, reducing stress and improving operational continuity.

Targeted information distribution is probably even more important in emergency and crisis communication than in service status reporting. BCM standards, e.g. BSI 200-4 [3] and [2] emphasize, that crisis communication is one of the primary success factors of crisis management. Crisis communication can be classified as internal and external communication and thus different stakeholders, interest and recipient groups must be identified and suitable communication channels established in advance. Effective crisis communication starts during the crisis itself and should not end abruptly once the crisis is believed to be over. Various stakeholders expect communication from the affected organization even after the crisis has passed. The primary goal of external communication is to provide target-group-specific information about the crisis, its causes, and how an organization deals with it. Communication aimed at internal target groups supports the work of the crisis management team and serves, e.g., to coordinate the specialist groups involved in crisis management, such as service administrators, the PR team, facility management, etc., or to obtain information from them. Other important questions that should be answered in advance of a crisis include, for example, who, i.e., which person or team is responsible for communicating with the respective target group, what information the target group expects and in what form, in particular the level of detail and language, i.e., more technical details or more of a management summary. A one-time communication to a target group is far from sufficient; regular updates are expected, so the frequency of such updates should also be considered. In our experience, when combined with service status reporting, these groups usually correspond to the customer or user groups of the respective service. The list may be extended to include, for example, the general public and institutional leadership.

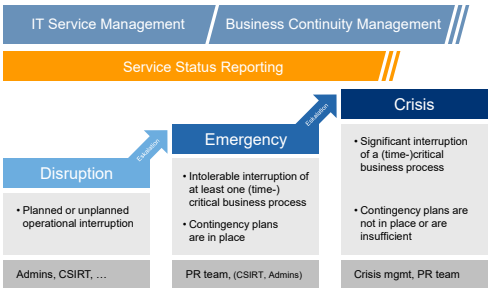

Figure 1: Service status reporting as a component in the handling of interruptions of increasing severity incl. authors at LRZ (based on BSI 200-4 [3] and [6])

The solution we present can contribute as a (technical) building block and organizational preparatory work for emergency and crisis communication, e.g. as described in detail in BSI 200-4 [3]. It is designed for unidirectional information transfer (service provider → user groups); it was thus set up as 'reporting'.[1] Fig. 1 outlines the role and contribution of status reporting within ITSM and BCM during service disruptions of varying severity.

[1] Any user feedback channels were considered secondary in emergencies and implemented via the existing ticket system (which is not guaranteed to be available in the event of a crisis). Establishing a resilient internal communication structure (e.g., direct messaging or voice communication within the CSIRT or crisis management team) is likewise advisable, but represents a separate use case not addressed here.

2.1 Requirements and Fundamental Design Decisions

Operational benefits such as fewer incident tickets for already known disruptions and higher customer satisfaction materialize only if reporting occurs more or less in real time. The LRZ therefore aims to publish notifications within 30 minutes of becoming aware of an incident. As [6] motivates, a status website was chosen over email as the reporting medium.[2]

At the LRZ, the focus was on the following use cases and value propositions of service status reporting and emergency communication, which are explained in more detail in [6]:

- Exactly one central overview of all services in the service catalog, including their current status – filterable for different internal and external stakeholder groups.
- Relief for the internal helpdesk / first-level support:
  - Fewer tickets reporting incidents that are already known
  - Enabling the helpdesk to respond to tickets by referring to the status board
- Timely communication targeted only at affected customer and user groups: Announcing maintenance work, reporting current incidents or service impairments, providing informative announcements (e.g., decommissioning dates for outdated client versions)
- Capability for external (inter-service) emergency and crisis communication, supporting CSIRT, crisis management team, and PR
- Publication and tracking of target-group-specific versions of announcements

The prevailing heterogeneity of the operator groups was successfully unified centrally by giving service owners autonomy in maintaining their respective sections of the status board, allowing them to tailor content, timing, and format to their user base. For example, the preferred language (e.g., english predominates in high-performance computing) and the required level of technical detail (e.g., services such as DNS or firewalls are typically not used by laypersons) vary accordingly. Before describing the implementation based on these requirements, it is necessary to define which "services" and content should be included in status reporting. The starting point for this process is Service Portfolio Management.

2.2 The Concept of a "Service" and IT Monitoring

As a service-oriented data center certified according to ISO/IEC 20000-1 [4], we define a service as a virtual package consisting an (IT) product, its operation, maintenance, further development, and support throughout its entire lifecycle. A service generally does not correspond to individual (technical) components but rather emerges as a combination of them. This understanding of a service is particularly practical for service status reporting. The subject of reporting is not whether, for example, a specific web server is currently operational, but whether the service it contributes to remains available and functional as a whole (possibly for only a subset of customers).

The basis for selecting the services displayed on the status board is the LRZ's service portfolio or service catalogue (DLK) and its concepts.[3].

[2]Status websites do not require laborious maintenance of mailing lists (per user sub-group), can be populated early and flexibly, updated at any time, and enriched with diverse content formats. Users can decide for themselves when and which (filtered) information they want to access (pull principle). However, there is no guarantee that all affected parties will see the information. Therefore, in time-critical cases, it may be useful to supplement this with the push-based via email (referring to the status board).

[3]See e.g. [7]

2.2.1 Granularity level of reporting

A granularity level must be selected for display on the status board. The LRZ service portfolio is structured hierarchically, with service classes to which individual services are mapped. The latter consist of service moduls and components maintained in the Configuration Management Database (CMDB). Fig. 2 illustrates this structure. Since most users generally lack (and do not require) technical knowledge of how their provider's services are built, reporting at the service level isgenerally a good starting point. Accordingly, all services from the service catalog that can have an operational-technical status were included, while purely advisory services were excluded.

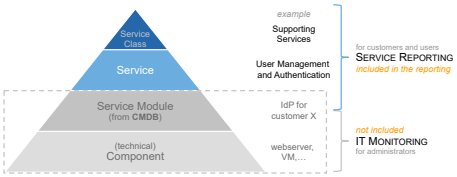

Figure 2: Granularity of listed services in the LRZ's service status reporting

2.2.2 From IT Monitoring to Service Status Reporting

Faults or interruptions occur at the level of service modules (or their subcomponents). However, due to redundancies or overcapacity, not every fault (e.g., a hard drive failure) results in a noticeable impact on service quality for users. Conversely, certain single failures, such as a severed network connection, can simultaneously affect multiple services.

Fully automated mapping of technical errors to service disruptions is complex: interdependencies among components, severity levels, affected user groups, and varying SLAs must be considered, requiring detailed modeling of service dependencies – a prerequisite often unmet. Pure IT monitoring with tools like *check_mk*, *Nagios* [4] etc. provides technical data but typically fall short for service status reports and rarely produces suitable texts for customers.

Given these challenges, small organizations or HEI generally benefit from a manual abstraction and validation layer for incident reports by the service owner. Once the service owner identifies sufficient indications of a disruption in their service (and possibly others) via IT monitoring etc., they issue a report considering affected users, root cause, known impacts, estimated resolution time, and, if applicable, guidance or alternative options for end users.

The proposed service status reporting concept grants each organization and its operational teams technical and organizational flexibility per service. This makes it adaptable to institutions of varying size and maturity, particularly in decentralized structures.[5] The following section outlines the technical implementation of this approach.

[4] https://checkmk.com/, https://www.nagios.org/

[5] A lightweight approach – starting without specialized tools – has proven effective at LRZ, including for document management within our integrated information security and IT service management system. [7]

3 Implementation of the Service Status Board

This chapter details the extended architecture of the status board at LRZ including its workflows. [6] described the selection and necessary adaptations of the initial product *cstate*[6] from over 100 evaluated candidates, as well as the basic architecture.

3.1 Architecture of the Status Board

The LRZ's status board's architecture is based on the static site generator *Hugo*[7]. Each message, i.e. page is created by the author automatically or (semi-)manually as a markdown text file including metadata such as its title and stored in a directory. Hugo then generates HTML pages from these files using predefined templates. cstate basically serves as a template ("theme") for Hugo.

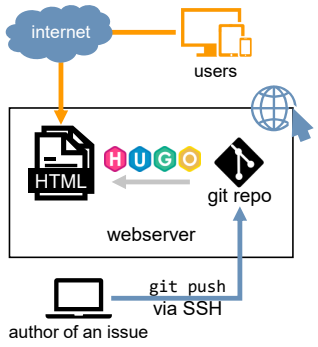

Figure 3: Generating static HTML pages with *Hugo* [6]

A graphical content management system (CMS) is also available for less technical authors to create and upload such text files. The status board follows the JAMstack architecture model. Instead of server-side (potentially dynamic) rendering on user request (involving database queries etc.) it delivers pre-rendered static HTML pages. This approach makes it relatively easy to archieve high performance, scalability, and maintainability.[8] Many security measures for database queries, server load peaks or user input validation, for example, become obsolete thanks to this approach. The template and markdown files are managed in a git repository, which also handles author authentication and authorization. Figure 3 illustrates the core architecture of a Hugo- or *cstate*-based website.

For production use, the architecture was expanded as shown in Fig. 4. To improve fault tolerance, the base setup was duplicated to $n = 2$ web servers. One server is located within the local university network, the Munich Scientific Network (MWN), and the second at a partner data center. Through DNS load balancing[9], users access the static HTML pages on either server via the uniform domain `https://status.lrz.de` . The web servers' git repositories are provisioned from a central GitLab repository, which also manages author authentication and authorization. User accounts for GitLab are provisioned from central directory services. The setup ensures traceability of changes and enables collaborative content creation. A GitLab pipeline deploys commits to both web servers via `rsync` after syntax and error checks, triggering Hugo to generate new static HTML pages. This process typically completes in under one minute. *Decap*[10] is used as a **content management system** for less technically savvy authors. Decap is stored as a JavaScript file on the web server and executed locally in the author's browser. As a headless CMS, Decap does not have its own data storage, but relies on the GitLab repo via API. Authentication and authorization for the CMS is done via OAuth2 against GitLab, i.e. with the author's regular credentials.

Entries in this graphical interface are committed to the GitLab repository as structured

[6] `https://github.com/cstate/cstate`
[7] `https://gohugo.io/`
[8] `https://jamstack.org/`
[9] [6] discusses the approach's fault tolerance and shows that, even in case of large scale outages, a – for LRZ – acceptable emergency operational level can be maintained.
[10] `https://decapcms.org/`

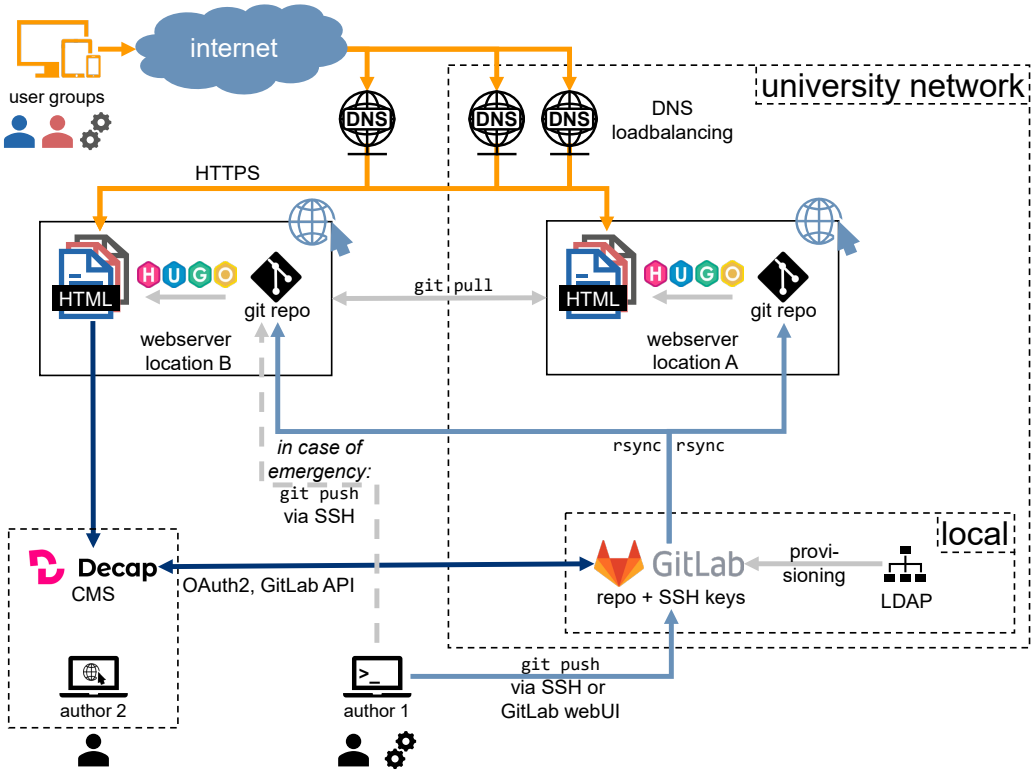

Figure 4: Architecture of the emergency and service status board, including its three paths for message creation. Colour coding of target group-specific HTML versions of the status board. (extends [6])

YAML- or markdown text files – exactly in the same format described in section 3.2, which can also be easily created manually. The CMS Decap thus acts as a git client, so to speak. Thanks to git, this offers the collaboration advantage of being able to create status messages on any channel (CMS, git command line, GitLab WebIDE etc.) and edit them via any other channel.

3.2 Service Status und Types of Messages

A consistent definition of service status (at least per service) is essential for service status reporting. At LRZ, the status board uses four states: Down, Disrupted, Maintenance, and Operational. As noted in section 2.2.2, automatically mapping technical faults or IT monitoring alerts to a service status is rarely straightforward. Service operator teams are therefore given flexibility to decide under which circumstances to set which status for their service. But they have to apply their chosen convention consistently within the service.

LRZ distinguishes two message types: incidents of varying severity (Down, Disrupted, or Maintenance) that affect service status, and purely informational announcements. A service's status is determined by the active message with the highest severity. [6] details the underlying data model.

A single message can affect multiple services, which is especially relevant for major disruptions or outages caused by inter-service dependencies. For both message types, a template is provided in the repository. Fig. 5 illustrates the file format.

The list of displayed services, along with their attributes such as description and links to documentation or the ticketing system, is maintained in a YAML config file.

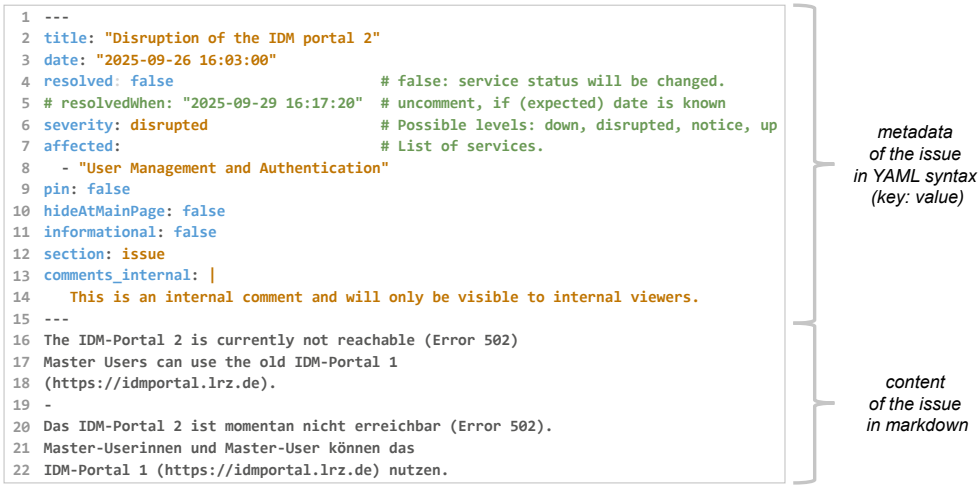

Figure 5: Text file of a simulated Incident with severity "Disruption" (`severity: disrupted`). Metadata in YAML-style; content in markdown.

3.3 Integrated target-group-specific Communication

To meet the requirements and benefits of the integrated delivery of tailored information to different recipient groups (see 2), multiple builds of the status board are necessary. In a static website approach, dynamic or viewer-session-dependent content is not possible.

Hugo generates HTML files by default in a `public/` directory. Additional builds can be triggered for each target group, differing only in the output directory and the value of a global variable. Depending on this variable, certain content elements are conditionally rendered in the layout template. The color coding of different HTML versions and user groups in Fig. 4 illustrates this. Hugo supports merging multiple configuration files, so a target-group-specific config file only needs to override the relevant global variable(s). The following commands show the builds of two page versions – for a public and a internal target group:

```
hugo --config config.yml --destination public/
hugo --config config.yml,config.internal.yml --destination internal/
```

To restrict access to the builds, it you can deploy the HTML files to separate web servers (e.g. one in the intranet of the target group) or implement proven methods of restricting access to a web server's subfolder, such as .htaccess or OAuth2 against GitLab, as is the case with the CMS. Depending on the implementation, not every one of these access control measures might be available to the specific target group in a crisis. As an emergency operating level the public build serves as a fallback.

To provide target-group-specific content, metadata attributes (i.e. hugo frontmatter) are added to the message template, as shown in Fig. 5, line 13-14. The markdown content of `comments_X` will be rendered only in the build for group `X` – below the public content. The public content thus will serve as the default, with the option to supplement it through additional target group specific sections beneath, thereby minimizing overall editorial effort.

3.4 Organisational Matters & Workflows

Workflows from the author's perspective To *publish* a message on the status board, authors (typically the affected service's admins) simply place a text file in the GitLab repository using the format shown in Fig. 5. This can be done manually, semi-automatically, or fully automatically – using templates in the repository. To *update* a message (e.g., add new information or mark it as resolved), the text file is edited; to *delete* a message, the file is removed.

Creating text files and committing them via git provides a simple interface that allowed LRZ to support all services efficiently. For example, when implementing some planned changes, the status board is automatically updated from the ticket tool, or corresponding announcements are published during their planning. Optionally, HTML files can be generated locally using the `hugo serve` command, allowing authors to review changes before publication. This also provides each author with a personal local testing environment.

Retrieving Information from the User's Perspective The status board is primarily accessed via the website. In addition, messages are available as RSS feeds (per service), which mail clients can display in an email-like format and filter using augmented attributes. This realizes benefits outlined in chapter 2. Additionaly, offering the content in JSON format supports integration into dashboards or digital signage displays, either on the customer side or within the LRZ operations center. Relevant messages can also be included in login portals or consoles of other LRZ services. This makes the status board the central source and leading system for service status updates. RSS and JSON files are generated by Hugo alongside the HTML pages whenever changes occur and are stored as static files on the web servers.

Use of target-group-specific Communication Using the same text file as a database for messages to different target groups (see 3.3) has advantages in terms of metadata consistency and collaborative updating of the respective information states. The incident's coordinator or PR team can always see which information has been shared with which audience. Thanks to git mechanisms, their history is automatically documented for post-incident review. As noted above, the same text file can be collaboratively edited through multiple channels (e.g., GitLab Web UI, CLI, or CMS). While admins keep technical data up to date, for example, the PR team maintains the sections relevant to their audiences via the CMS.

This workflow is fully integrated into routine service status reporting. If necessary, e.g. in emergencies, additional or tailored information can be provided to specific target groups, while the base version containing only public information remains available in all builds.

Contingency procedure for publishing notifications A "contingency procedure" was developed to guard against failures in the publication workflow: if the central GitLab repository becomes unavailable (e.g., due to storage, LDAP, or network… outages), messages can be pushed directly to the git repositories of the web server(s) that are still reachable. Standard git collaboration features remain available; the CMS is not.

Administrators' SSH keys, already stored in GitLab, are synchronized to the web servers' `authorized_keys`. The repository's `README.md` contains full user documentation, including the addresses of the external web server and the contingency procedure. This is therefore automatically available to every message author from the first message via the git command line in their local copy of the repo. Once GitLab is restored, changes made directly on the web servers are overwritten by the next commit to GitLab. Any updates made on the webservers are still locally and can, if still relevant, simply be pushed to the GitLab then. Even in the

event of widespread outages of central services, one of the main publication channels (git via CLI) remains operational: a simple `git push` to an alternate target address ensures continuity.

3.5 Look and Feel of the Status Board

The homepage provides an up-to-date overview of all service classes and their respective services with current status (Fig. 6). Direct links to each service's ticketing system and user documentation are easily accessible. Each service has its own page listing all related messages (Fig. 7). Direct links to individual messages can be shared by the LRZ's Service Desk or the admin team with interested parties. RSS subscriptions are highlighted at the bottom of each page. The simulated incident message from Fig. 5, can later be edited via the CMS (see Fig. 8) and, as illustrated in Fig. 9, is also optimized for mobile devices. Messages with broad impact or for cross-service emergency and crisis communication can be pinned at the top.

4 Discussion, Conclusion and Outlook

With the status board conceptualized in chapter 2 and implemented in chapter 3, Leibniz-Rechenzentrum established a technical solution to integrate service status reporting with unidirectional (external) communication in crisis situations. This created a common, integrated building block for ITSM, ISM, and BCM that meets the requirements defined in chapter 2.1. However, the respective communication strategies, qualitative content and target groups must be developed in the fields of ITSM and BCM.

A deliberately lean technical approach was chosen for the status board as a centralised Service Status Reporting system. Implementation is based on static HTML pages, versioned via git and based on established administration tools. After nearly three years of operation, this reduction to simple, robust components has proven to deliver a low-maintenance, fail-safe, and scalable solution.

The decision to use static content increases stability and resilience but limits dynamic or date-dependent functions. Such logic must be implemented outside the status board (e.g., via scripts or gitlab pipelines). The board itself does not perform monitoring; it serves "merley" as a communication instrument.

In day-to-day operations, the advantages clearly prevail. The status board presents information in a structured manner at the service catalog level, enhancing transparency for users. The established announcement workflow enables early maintenance notices, updates during incidents, and final resolution messages. Despite heterogenity in LRZ's internal working methods, service architectures, and maturity levels, users are provided with a unified status reporting interface. RSS feeds, filtering options, and target-group-specific versions support targeted information in near real time. Further operational experience is discussed in detail in [6].

For authors, the solution proved to be easily integrated into existing processes. Messages can be generated manually or, for example, scripted from tickets. The use of common admin tools lowered entry barriers and fostered acceptance. The Service Desk benefits in particular: it has has an up-to-date, service-oriented overview and can respond to inquiries by referring to the status board. In the event of extensive disruptions, reports can be updated centrally for multiple target groups, reducing redundant communication channels and repeated status inquiries.

From an organisational perspective, introducing centralized service status reporting was challenging. Previously decentralized solutions had to be consolidated, and numerous stakeholders involved. Acceptance was facilitated by familiar tools and flexible workflows. At the

same time, the public visibility of content required raising authors' awareness of wording, level of detail, and tone. Feedback processes proved useful without introducing rigid approval structures.

Fundamentally, the status board primarily addresses the "how" of communication – channels, structure, and technical integration. The "what," including content quality, target groups definition, and communication strategy, remains the responsibility of IT Service Management and Business Continuity Management. Especially in crises, preparation is essential: predefined guidelines or text modules help ensure consistent, target-group-specific information under time pressure. Underlying organizational processes and concepts are critical. For service status reporting that goes beyond pure emergency communication, clear definition of the service concept and the service portfolio, as well as the systematic (recording and) linking of service modules and their components, are particularly relevant. Introducing a status board with visible added value for users and admins can provide a good opportunity for this clarification.

The central status board has made a lasting contribution at Leibniz Supercomputing Centre to professionalisation of service status communication – not only in emergencies. It enhances transparency and trust, relieves the service desk and operations teams, and improves responsiveness during incidents and crises. Originating from BCM and ISM initiatives, its introduction also laid a useful foundation for strengthening ITSM. Overall, the status board represents a field-tested and transferable approach and technical architecture for service status and external communication in emergency and crisis scenarios – an area, for which the higher education environment is increasingly preparing.

References

[1] Axelos. *ITIL4: Create, Deliver and Support*. ITIL4 Managing Professional. Stationery Office, 2020.

[2] Bundesamt für Sicherheit in der Informationstechnik. 100-4. notfallmanagement.

[3] Bundesamt für Sicherheit in der Informationstechnik. 200-4, business continuity management.

[4] International Organization for Standardization [ISO]. Iso/iec 27001:2013. information security management systems – requirements.

[5] International Organization for Standardization [ISO]. Iso/iec 27001:2022. information security, cybersecurity and privacy protection — information security management systems – requirements.

[6] Miran Maximilian Mizani. Robuste notfall- und service-status-kommunikation – konzeption und realisierung eines hochverfügbaren service-statusboards als beitrag zu bcm, ism und itsm. In *Sicherheit in vernetzten Systemen: 33. DFN-Konferenz*, pages E–1 – E–30. epubli (Neopubli GmbH), 2026.

[7] Michael Schmidt, Stefan Metzger, and Miran Mizani. Managementsysteme ohne spezialisierte tools etablieren: Ein leichtgewichtiger ansatz zur dokumentation im service- und informationssicherheitsmanagement. In A. Ude, editor, *Sicherheit in vernetzten Systemen: 29. DFN-Konferenz*, pages A–1 – A–20, Hamburg, Januar 2022. Books on Demand.

A Web Interface of LRZ's Status board

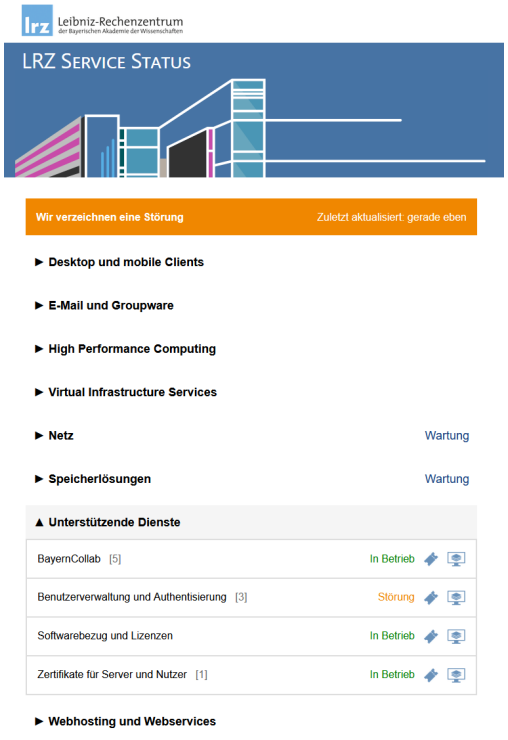

Figure 6: Excerpt of the status board showing a simulated incident; service class "Supporting Services" expanded (see [6])

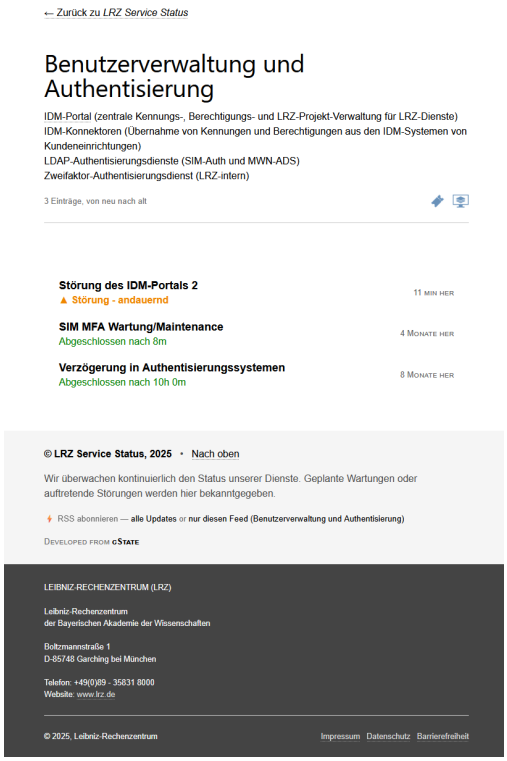

Figure 7: Message overview for the service "User Management and Authentication" with a simulated incident (see [6])

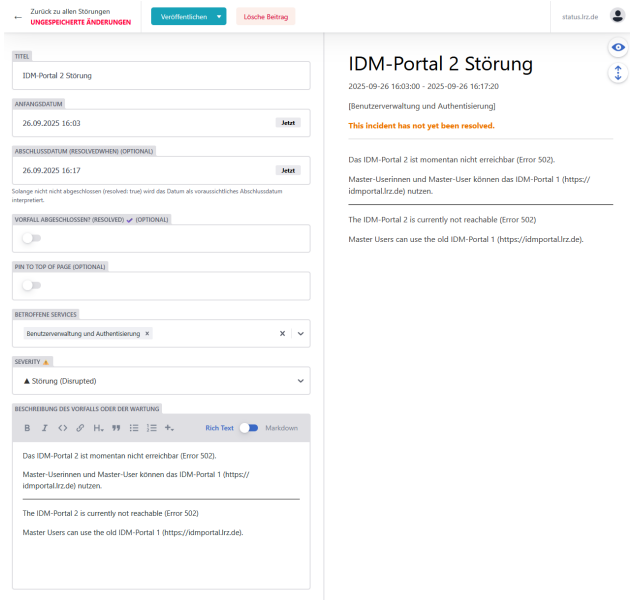

Figure 8: Simulated incident message in the CMS with integrated preview functionality (useful for complex markdown elements, such as tables) (see [6])

Figure 9: Individual view of a simulated incident report with internal comments displayed. Affected services are annotated. (see [6])

B Acknowledgments and Declarations

This article extends our previous work [6] presented at the german "33. DFN-Konferenz: Sicherheit in vernetzten Systemen". While the prior publication introduced the architecture and initial implementation, this paper substantially refines and extends the approach. We have streamlined the presentation while retaining relevant elements, incorporated the overwhelmingly positive feedback from early practical adopters and introduced the concept of target-group–specific communication channels integrated into routine service status reporting – further supporting ITSM, ISM and BCM. This results in a more versatile and feature-rich setup that is transferable to other higher education institutions.

Use of AI The original content of this paper was created by the authors without any generative AI. AI tools were used for editorial purposes, such as spelling, language and grammar checking.

C Author Biographies

Mizani, Miran Maximilian Miran Mizani studied Philosophy, Biology and Computer Science at Ludwig Maximilians University (LMU) Munich as well as Technology Management at the Center for Digital Technology and Management (CDTM). Since 2020, he has been acting as IT Security Architect at Leibniz Supercomputing Centre (LRZ) while pursuing his Ph.D at LMU. He focuses on information security management in a inter-departmental and inter-organizational context. https://orcid.org/0000-0002-7307-6757

Metzger, Stefan Stefan Metzger studied Computer Science at Technical University of Munich (TUM). In 2009, he joined the LRZ as a security analyst and spent several years working on a European research project, including in the field of security. Since 2017 he has held the position of Chief Information Security Officer and is responsible for the information security management system at the LRZ. His research focus is on information security in inter-organizational environments. https://orcid.org/0009-0002-4753-3659